REMARKS

In paragraph 3 of the Office action, the examiner states that the Information Disclosure Statement filed 10/20/2003 did not contain the required legible copy of each non-patent literature publication. In response, a copy of form PTO/SB/08B and copies of the two publications listed on that form are filed herewith.

Paragraph [0083] of the specification has been amended to reflect the serial number and filing date of the application identified therein.

35 U.S.C. § 101

In paragraph 5 of the Office action, claims 1 and 20 stand rejected under 35 U.S.C. § 101 because the claims allegedly do not "produce any tangible result . . . at the end of the process" and "the steps are directed to a computer program per se representing functional descriptive material." Claim 1 has been amended to recite that the balancing is repeated "until the work load is balanced across all of said processing elements." That amendment provides a tangible, real-world result at the end of the process. Claim 1 has also been amended to recite that the balancing is accomplished by "redistributing tasks amongst the processing elements in said line." The redistribution of tasks is a physical act requiring the transfer of tasks from one processing element to another. Such physical acts demonstrate that claim 1 is not a computer program per se.

Claim 20 has been amended to recite that the goal of the method is to balance one dimension of an n-dimensional array of processing elements. Claim 20 has also been amended to recite that the balancing, e.g., the goal, is accomplished by redistributing tasks amongst the processing elements in each of said plurality of said lines. The redistribution of tasks is a physical act requiring the transfer of tasks from one processing element to another. Such physical acts demonstrate that claim 20 is not a computer program per se.

In paragraph 5 of the Office action, claim 26 stands rejected under 35 U.S.C. § 101 because the claim recites "a memory device." In response, claim 26 has been amended to recite "a computer readable memory device." Claims to a "computer readable medium" are authorized in the Interim Guidelines for Subject Matter Eligibility, in the section dealing with "practical application." It is believed that claims 1, 20, and 26, as amended, set forth statutory subject matter such that the 35 U.S.C. § 101 rejection should be withdrawn.

35 U.S.C. § 112

In paragraph 8 of the Office action, claims 1-33 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

In paragraph 8ai, the examiner states that claims 1 and 33 recite in lines 5-6 "balancing at least one line of processing elements in the first dimension; balancing at least one line of processing elements in a next dimension; it is not clear how the balancing is being done and how the dimension is being defined." Claims 1 and 33 have been amended to make it clear that the balancing is being done by redistributing tasks amongst the processors in the line. With respect to the allegation that it is not clear how the dimension is being defined, the examiner's attention is respectfully directed to the description of a preferred embodiment appearing in the specification. More specifically, paragraph [0060] of the application as filed provides:

> After the lines in the first dimension are balanced (e.g., the rows), the next dimension (e.g., the columns) is balanced in the simple load balancing method as described in conjunction with FIG. 4 according to an embodiment of the present invention.

It is respectfully submitted that a person of ordinary skill in the art would be well aware of what is meant by "a first dimension" and "a next dimension." If the examiner does not agree, the examiner is invited to suggest language which the examiner believes will address the examiner's concern.

In paragraph 8aii, the examiner states, with respect to claim 20, that is unclear how lines are balanced to be having only values of X and X+1 and that the language the "sum of processing elements relative to a second dimension has two values" is also unclear. First, claim 20 has been amended to recite that balancing is accomplished by redistributing tasks amongst the processing elements in the line. As discussed in the specification as filed, paragraphs [0066], [0067], and [0068] provide:

> As is apparent in FIG. 6, each row of the row-balanced array 50a has either X or (X+1) tasks associated with each PE (it should be noted that the value of X may

> be different for each row in the row-balanced array). In the general method of the current embodiment, the values 0 and 1 are substituted for X and (X+1), respectively. For example referring to the first row of array 50a in FIG. 6, it is apparent that X is equal to five (5) and (X+1) is equal to six (6). Likewise for the second row, X is equal to four (4) and (X+1) is equal to five (5). Thus as seen in the first row of array 50e, a zero (0) is substituted for all PEs having five (5) (i.e., X) tasks and one (1) is substituted for all PEs having six (6) (i.e., X+1) tasks and for the second row a zero (0) is substituted for all PEs having four (4) (i.e., X) tasks and one (1) is substituted for all PEs having five (5) (i.e., X+1) tasks. Likewise, a zero or one is substituted for each row of array 50a. The substitutions are completed in parallel for all rows of the array. The result of the substitution is illustrated in array 50e.
>
> Summing the tasks on each column of array 50e, it is apparent that the columns range from zero to seven tasks per column. It should be noted that the column sums represent the different rounding errors that are incorporated into the column sums. To create an optimal load balance, it is desirable to have only two different rounding errors at the end of each dimension stage (e.g., row, column, etc.)
>
> To limit the rounding error to two values, one embodiment of the general method of the current embodiment employs a shifting technique. Referring to arrays 50e and 50f, the first row of array 50e is not shifted. The second row down is shifted to the left until the rightmost one (1) of the second row is under the rightmost zero (0) of the row directly above (i.e., the first row). The third row down is shifted left until the rightmost one (1) of the third row is under the rightmost zero (0) of the row directly above (i.e., the second row). Each subsequent row is treated in the same manner. The effect is to create an irregular staircase of ones (1) (as illustrated by the dark lines in array 50f). Any data that "falls off" the left hand edge of the row is wrapped around onto the right hand edge of the same row. If the rows are shifted as discussed and the columns summed as shown in array 50f, the rounding errors can be limited to two values (i.e., 3 and 4).

It is respectfully submitted that when the language of the claim is read in the context of the three paragraphs reproduced above, and the example shown in figure 6 discussed in those paragraphs is reviewed, the language of the claim is clear.

In paragraph 8aiii, the examiner states that in claims 2 and 22 "calculating a local mean number of tasks within each of said plurality of processing elements" is unclear. Applicant respectfully disagrees. The claims are read in light of the specification, and the specification discloses at least one method of calculating a local mean. The application as filed in paragraph [0083] provides:

> After the total number of tasks (V) present on the first row is distributed in operation 62, the local mean number (M_r) of tasks for each PE_r in the row is calculated in operation 63. In the current embodiment, the local mean value is computed using the rounding function $M_r = Trunc((V + E_r)/N)$ (where M_r represents the local mean for PE_r, N represents the total number of PEs 30 in the row, and E_r represents a number in the range of 0 to $(N-1)$, as derived in conjunction with the general method illustrated in Table #1 and Table #2), to ensure that no instructions are lost or "gained" during the rounding process if the value of $V \div N$ is not an integer (i.e., to ensure that $V = \sum_{i=0}^{i=N-1} M_i$, where N represents the number of PEs 30 in the row, and M_i represents the local mean of tasks associated with a local PE_r in the row). The rounding function is discussed in more detail in U.S. Patent Application Serial No. 10/689,382 entitled "Method for Rounding Values for a Plurality of Parallel Processing Elements" filed Ocotber 20, 2003 and incorporated in its entirety by reference herein.

Applicant asserts that one of ordinary skill in the art would know how to calculate a local mean based on the disclosure in the specification. Applicant should not be required to write a preferred embodiment into the claims.

In paragraph 8aiii, the examiner next states that in line 11 it is unclear whether the local deviation determining step is performed based on the preceding step. Claim 2 has been amended to make it clear that the local deviation is calculated from the local mean number. Support for that change can be found in paragraph [0088] which provides "After the local means (M_r) are computed in operation 63, the local deviation D_r is calculated for each PE_r in the line in operation 64. In the current embodiment, the local deviation is simply the difference between the local value and the local mean (i.e., $D_r = v_r - M_r$)." Claim 22 has been amended to address the examiner's concern. Support for the amendment to claim 22 can be found in the application as filed in paragraphs [0088]-[0091].

In paragraph 8aiv, it is the examiner's position that it is unclear what is meant by the "V" in claims 7 and 25. Each of claims 7 and 25 has been amended to recite that "V" is the total number of tasks.

In paragraph 8aiv, the examiner also indicates, with respect to "E_r" in claims 5 and 18, that it is unclear how that value is determined for each of the plurality of processing elements. The examiner's attention is respectfully directed to paragraph [0085] of the application as filed which provides:

> The rounding function $M_r = Trunc((V + E_r)/N)$ prevents tasks from being lost or gained. In the current embodiment, each PE is assigned a different E_r value for controlling the rounding. The simplest form for the function E is the case in which $E_r = P_r$, where P_r represents the PEs position in the row. For example, for PE_0, $E_0 = 0$; for PE_1, $E_1 = 1$; for PE_2, $E_2 = 2$; etc. By assigning each PE in the row a different E_r value, the rounding function can be controlled such that some of the local means are rounded up and some of the local means are rounded down, thus insuring that $V = \sum_{i=0}^{i=N-1} M_i$. It should be noted that in the current embodiment, the local mean for each PE 30 in the row is computed in parallel with the local means of the other PEs in the row. It should further be noted that the local mean for PEs in all the rows of the array are computed in parallel.

It is submitted that reading claims 5 and 18 in view of the disclosure of paragraph [0085], one of ordinary skill in the art would understand how the value E_r is derived for each of the plurality of processing elements.

In paragraph 8aiv, the examiner indicates with respect to claims 7 and 25, that no definition is provided for PE_r. Claims 7 and 25 have been amended to provide a definition for PE_r.

In paragraph 8av, the examiner indicates that it is unclear in claims 9 and 26 how E_r "controls" the *Trunc* function. The language of claim 9 and claim 26 has been amended to recite that the *Trunc* function is responsive to the value of E_r. With respect to the examiner's question about how this step is possible, "since each E_r value is set ahead of time and must be different for each processing element," the examiner's attention is respectfully directed to paragraph [0085] reproduced above.

With respect to paragraph 8avi, the examiner states that the recitation of "X and (X+1)" in claims 10 and 27 is unclear. The examiner's attention is respectfully directed to paragraph [0016] of the application as filed which provides as follows:

> The present invention enables tasks to be distributed along a group of serially connected PEs so that each PE typically has X number of tasks or (X+1) number of tasks to perform in the next phase. The present invention may be performed using the hardware and software (i.e., the local processing capability) of each PE within the array. Those advantages.

The examiner's attention is also directed to the table appearing in paragraph [0086] of the application as filed which provides:

v_r	E_r	$(V+E_r)/N$	$M_r = Trunc((V+E_r)/N)$	D_r
2	0	5.125	5	-3
4	1	5.25	5	-1
6	2	5.375	5	1
8	3	5.5	5	3
3	4	5.625	5	-2
4	5	5.75	5	-1
6	6	5.875	5	1
8	7	6	6	2

Table #3 – Local Mean Calculation for the First Row of Array 50 ($V = 41, N = 8$).

The language of claims 10 and 27 has been amended to recite that a local mean for each group is equal to either X, or X+1, as seen clearly from Table No. 3 where X = 5 and X+1 = 6.

A definition for E_r can be found in claim 7, from which claim 10 depends. Claim 27 has been amended to depend from claim 25, which contains a definition for E_r.

In view of the foregoing, it is respectfully requested that the rejection of claims 1-33 under 35 U.S.C. § 112, second paragraph, be withdrawn.

35 U.S.C. § 102

In paragraph 10 of the Office action, claims 1, 18, and 33 stand rejected under 35 U.S.C. § 102 as being anticipated by U.S. Patent No. 5,630,129 (Wheat). It is respectfully submitted that the examiner reads too much into Wheat.

The examiner is correct in asserting that Wheat discloses a method of load balancing, but the method disclosed in Wheat is very different from the subject matter of claims 1 and 33. Claim 18 has been cancelled as the substance of claim 18 is now found in amended claim 1. The portion of Wheat cited by the examiner provides:

> The present invention is of a method and apparatus for dynamically maintaining global load balance on a parallel computer, comprising: providing an application for execution to a plurality of processors of the parallel computer, the application comprising a plurality of data cells arranged spatially such that each data cell has one or more neighboring data cells; assigning each data cell to a processor; determining for one or more processors all other processors in corresponding processor neighborhoods; computing work loads for one or more processors; and

> for one or more processors, exporting one or more data cells to another processor in a corresponding processor neighborhood. In the preferred embodiment, the determining, computing and exporting tasks are performed repeatedly until the application has completed execution, and global load imbalance is minimized within a finite number of iterations. (Emphasis added.)

The actual balancing method of Wheat is set forth beginning in column 5, line 50, with a determination of workloads. Workloads are then compared amongst processors. See column 5, lines 60-67, which provide:

> Each processor compares its work load to the work load of the other processors in its neighborhood and determines which processors have greater work loads than its own. If any are found, it selects the one with the greatest work load (ties are broken arbitrarily) and sends a request for work to that processor. Each processor may send only one work request, but a single processor may receive several work requests.

Transfers take place according to priorities as discussed in column 6, lines 40-57, which provide as follows:

> FIG. 4 illustrates an example of element priorities and selection for exporting four elements to the east neighboring processor. Initially, elements 3, 6, 9, and 12 are eligible for export. Their priorities are computed; element 3, for example, has priority -2, since it has two local neighbors (-2), one neighbor in a concerned partner processor (-2), and one neighbor in the importing processor (+2). Elements 6 and 9 share the highest priority, but since element 6 has a greater work load, it is selected. Element 5 becomes eligible for export, but its priority is low since it has three local neighbors. The priorities are adjusted, and element 9 is selected, making element 8 a candidate. The priorities are again updated, and the selection process continues with elements 3 and then 12 being selected. Although the work request is not completely satisfied, no other elements are exported, as the work loads of the elements with the highest priority, 5 and 8, are greater than the remaining work request.

It is seen that Wheat, although disclosing a method for dynamic load balancing, teaches a very different method from what is claimed in claims 1 and 33. Processor work requests are determined based on processors comparing their workloads with other processors. Requests are then made and granted on the basis of priorities. There is no redistributing tasks amongst the processors in a line in a first dimension, no redistributing tasks amongst the processors in a line in a next dimension, and no repeating until the load is balanced across all of the processing elements. It is respectfully submitted that the rejection of claims 1 and 33 under 35 U.S.C. § 102 as being anticipated by Wheat be withdrawn.

35 U.S.C. § 103

In paragraph 12 of the Office action, claims 2, 3, 4. 6, 11, 16, 17, 19-24, and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Wheat in view of "A Simple Load Balancing Scheme for Task Allocation in Parallel Machines" (Rudolph).

Per claim 2, it is the examiner's position that Wheat discloses "calculating a total number of tasks for said line, wherein said total number of tasks for said line equals the sum of said local number of tasks for each processing element[s] on said line" citing column 5, lines 52-56. That portion of Wheat discloses:

> Each processor determines its work load as the time to process its local data since the previous balancing phase less the time to exchange inter-processor boundary data during the computation phase. Neighborhood average work loads are also calculated.

There is no mention of calculating a total number of tasks for a line as asserted by the examiner. That is because Wheat does not disclose the balancing of lines of processing elements.

The examiner also cites column 5, lines 55-56, for a disclosure of "calculating a local mean number of tasks for each processing element[s] on said line." Again, there is no mention of calculating anything based on a line of processing elements as Wheat does not disclose the balancing of lines of processing elements.

The examiner next asserts that "determining a first local cumulative deviation for each of said plurality of processing elements" and "determining a second local cumulative deviation for each of said plurality of processing elements" is taught at column 12, lines 59-66. The examiner, however, has not quoted the entire limitation from the claim. In their entireties, the limitations read "determining a first local cumulative deviation for each of said plurality of processing elements <u>on said line</u>" and "determining a second local cumulative deviation for each of said plurality of processing elements <u>on said line.</u>" The portion of Wheat cited by the examiner provides:

> FIGS. 9 and 10 illustrate the convergence of the processor work loads from uniform domain decomposition toward global balance. FIG. 9 shows the +1 and -1 standard deviation curves of the maximum computation time for each time step. Initially, the deviation is large, indicating the processors are far from global balance. The deviations quickly become smaller, indicating the processors rapidly approach balance.

The cited portion of Wheat does not support the examiner's position. It is thus seen that Wheat does not disclose those portions of claim 2 as asserted by the examiner. For that reason alone, the rejection of claim 2 under 35 U.S.C. § 103(a) should be withdrawn.

Turning next to Rudolph, it is noted that in all of the citations of language from claim 2, the examiner has left out the following underlined language:

"notifying each of said plurality of processing elements of said total number of tasks for said line;"

"calculating a local deviation for each of said plurality of processing elements on said line."

When the underlined language is taken into consideration, it is seen that Rudolph falls short of supplying the necessary disclosure.

It is also the examiner's position that Rudolph discloses "redistributing tasks among said plurality of processing elements in response to said first local cumulative deviation and said second local cumulative deviation." The examiner cites page 3, column 1, figure 1, and column 1, lines 19-22. While it is true that Rudolph teaches redistribution of tasks between processors, the redistribution is not responsive to a first and second local cumulative deviation. That is made clear in Rudolph, second full paragraph on page 4 which recites:

> The load-balancing task simply chooses some other PE at random and tries to equalize the load between the two workpiles (see Figure 2). If the difference in load between the two workpiles is greater than some lower limit, tasks are then migrated from the heavier loaded workpile to the lighter one. If the other workpile is currently being accessed, then either the PE may give up or else wait until the workpile becomes free. Our implementations suggest that there is little difference between these strategies. (Emphasis added.)

It is thus seen that Rudolph does not disclose those portions of claim 2 as asserted by the examiner. For the foregoing reasons, it is respectfully submitted that the rejection of claim 2 under 35 U.S.C. § 103(a) should be withdrawn. Note that claim 2 has been amended to correct a grammatical error (viz. "each processing elements" has been changed to - - each <u>of said</u> processing elements - - at several locations).

As per claim 20, the examiner asserts that Rudolph discloses the substance of this claim at page 4, column 1, lines 37-44. The examiner states that the "system threshold value τ is the value of X and values more that *[sic than]* τ is X+1. System is being balanced according to the τ value means shifting the task loads." The examiner's statement ignores several limitations in claim 20. The balancing in claim 20 is performed by redistributing tasks amongst the processing elements in a line. The examiner has pointed to no such teaching in Rudolph. The shifting in the claim is a shifting of values within a line. The examiner has pointed to no such teaching in Rudolph. The shifting produces a sum that has only two values relative to a second dimension. Again, the examiner has pointed to no such teaching in Rudolph. For those reasons, the rejection of claim 20 under 35 U.S.C. § 103(a) should be withdrawn.

The remainder of the dependent claims not specifically argued in this amendment contain all of the limitations recited in their base claims. Because all of the base claims are believed to be in condition for allowance for the reasons set forth above, all of the remaining dependent claims are also believed to be in condition for allowance. Applicant reserves the right to argue the patentability of the dependent claims separately, at a later date, should that become necessary.

<u>Request for Interview</u>

Applicant has made a diligent effort to place the instant application in condition for allowance. If the examiner is of the opinion that the instant amendment does not place the currently pending claims in condition for allowance with respect to the art of record, the examiner is respectfully requested to contact applicant's attorney at the telephone number listed below **so that an interview may be scheduled before the issuance of a final Office action rejecting the claims on the basis of the art currently of record.**

Furthermore, the amendments to claims 1, 20, and 33 incorporate the subject matter of cancelled claim 18 and/or otherwise make clear what was previously inherent. As a result, the present amendment does not necessitate a new search and therefore cannot be the basis for a final Office action rejection based on a new ground of rejection.

Finally, the undersigned attorney wishes to draw the examiner's attention to the related applications listed in the first paragraph of the instant application. Several of those applications are related to load balancing and all have generated double patenting rejections. None of those double patenting rejections, however, involves the instant application.

Respectfully submitted,

Edward L. Pencoske
Reg. No. 29,688
Jones Day
One Mellon Center
500 Grant Street, Suite 3100
Pittsburgh, PA, USA, 15219
(412) 394-9531
(412) 394-7959 (Fax)
Attorneys for Applicant